

Cracking a hierarchical chaotic image encryption algorithm based on permutation

Chengqing Li*

College of Information Engineering, Xiangtan University, Xiangtan 411105, Hunan, China

Abstract

In year 2000, an efficient hierarchical chaotic image encryption (HCIE) algorithm was proposed, which divides a plain-image of size $M \times N$ with T possible value levels into K blocks of the same size and then operates position permutation on two levels: intra-block and inter-block. As a typical position permutation-only encryption algorithm, it has received intensive attention. The present paper analyzes specific security performance of HCIE against ciphertext-only attack and known/chosen-plaintext attack. It is found that only $O(\lceil \log_T(M \cdot N/K) \rceil)$ known/chosen plain-images are sufficient to achieve a good performance, and the computational complexity is $O(M \cdot N \cdot \lceil \log_T(M \cdot N/K) \rceil)$, which effectively demonstrates that hierarchical permutation-only image encryption algorithms are less secure than normal (i.e., non-hierarchical) ones. Detailed experiment results are given to verify the feasibility of the known-plaintext attack. In addition, it is pointed out that the security of HCIE against ciphertext-only attack was much overestimated.

Keywords: Chosen-plaintext attack, chaotic cryptanalysis, known-plaintext attack, permutation.

1. Introduction

With the increasing transmission speeds of wired/wireless networks and popularization of image capturing devices and cloud storage services, image data are transmitted over open networks more and more frequently. This makes security of image data become more and more important. The public concern of it becomes serious as news about the illegal online leak of personal photos of some celebrities was released. As a chaotic system owns some similar properties as that of modern encryption schemes, it has been intensively studied as an alternative approach for designing secure and efficient encryption schemes [1, 2, 3]. The main idea and principle of applying chaos theory to protecting images can be traced back to 1986 [4], which demonstrates the stretching effect of a chaotic map on a painting of Henri Poincaré, a founder of modern chaos theory.

The simplest and most efficient method for protecting multimedia data is permuting the positions of their spatial pixels [5] or frequency coefficients [6]. In the literature, some synonyms of permutation, transposition, shuffle, scramble [6], swap and shift, are used. Security scrutiny on some specific permutation-only encryption algorithms

against known/chosen-plaintext attacks were previously developed [7, 8]. In [9], a ciphertext-only attack on a specially simple permutation-only encryption algorithm was proposed utilizing correlation redundancy remaining in the cipher-image. No matter how the permutation relationships are generated and what the permutation object is, any permutation-only encryption algorithm can always be represented by a *permutation relationship matrix*, whose entry stores the corresponding permuted location in the ciphertext [10]. The security of permutation-only encryption algorithm relies on its *real permutation domain*, in which any element in the permutation object can be permuted independently. As for a permutation domain of size $M \times N$ with T possible value levels, it is estimated that the required number of known/chosen-plaintexts for an efficient plaintext attack is $O(\lceil \log_T(M \cdot N) \rceil)$, where $\lceil x \rceil$ denotes the ceiling function. An upper bound of the attack complexity is also derived therein to be $O(n \cdot (M \cdot N)^2)$, where n is the number of known/chosen plain-images [10]. In [11], the computational complexity of the attack is further reduced to $O(n \cdot (M \cdot N))$ by replacing the set intersection operations of quadratic complexity with linear element access operations. Even so, all kinds of permutation operations are still being used in multimedia protection today [12, 13, 14, 15].

In [16], a typical example of permutation-only image encryption algorithms, called HCIE (hierarchical chaotic image encryption), was proposed. Although security perfor-

*Corresponding author.

Email address: DrChengqingLi@gmail.com (Chengqing Li)

mance of general permutation-only image encryption algorithms against plaintext attack has been quantitatively analyzed, specific security performance of HCIE is still not evaluated. The core of HCIE is a permutation function composed of rotation operations of four directions, originates from an intellectual toy, Rubik's Cube [17]. In [16], the authors claimed about the security property of HCIE as follows: "By way of collecting some original images and their encryption results or collecting some specified images and their corresponding encryption results, it is still difficult for the cryptanalysts to decrypt an encrypted image correctly because the permutation relationship is different for each image." In this paper, we will demonstrate that the claim on the robustness of HCIE against known/chosen-plaintext attack is groundless. Further more, we find that the hierarchical encryption structure suggested in HCIE does not provide any higher security against known/chosen-plaintext attack, but actually make the overall security even weaker. In addition, we find the capability of HCIE against ciphertext-only attack was much over-estimated.

The rest of this paper is organized as follows. The algorithm HCIE is briefly introduced in Sec. 2. Detailed cryptanalysis on HCIE is provided in Sec. 3, with some experimental results. The last section concludes the paper.

2. The hierarchical chaotic image encryption algorithm (HCIE)

HCIE is a two-level hierarchical permutation-only image encryption algorithm, in which all involved permutation relationships are defined by pseudo-random combinations of four rotation mappings with pseudo-random parameters. For an image, $f = [f(i, j)]_{M \times N}$, the four mapping operations are described as follows, where $p < \min(M, N)$ holds for each mapping.

Definition 1. The mapping $f' = \text{ROLR}_b^{i,p}(f)$ ($0 \leq i \leq M - 1$) is defined to rotate the i -th row of f , in the left (when $b = 0$) or right (when $b = 1$) direction, by p pixels.

Definition 2. The mapping $f' = \text{ROUD}_b^{j,p}(f)$ ($0 \leq j \leq N - 1$) is defined to rotate the j -th column of f , in the up (when $b = 0$) or down (when $b = 1$) direction, by p pixels.

Definition 3. The mapping $f' = \text{ROUR}_b^{k,p}(f)$ ($0 \leq k \leq M + N - 2$) is defined to rotate all pixels satisfying $i + j = k$, in the lower-left (when $b = 0$) or upper-right (when $b = 1$) direction, by p pixels.

Definition 4. The mapping $f' = \text{ROUL}_b^{l,p}(f)$ ($1 - N \leq l \leq M - 1$) is defined to rotate all pixels satisfying $i - j = l$, in the upper-left (when $b = 0$) or lower-right (when $b = 1$) direction, by p pixels.

Given a pseudo-random bit sequence $\{b(i)\}$ starting from i_0 , the Sub_HCIE function in Algorithm 1 is used to permute an $S_M \times S_N$ image f_{sub} to become another $S_M \times S_N$ image f'_{sub} , where $(\alpha, \beta, \gamma, no)$ are control parameters. One can see

Algorithm 1 The Sub_HCIE function

```

1: function SUB_HCIE( $f_{sub}, \{b(i)\}, no, S_M, S_N$ )
2:   for  $ite \leftarrow 0, no$  do
3:      $q \leftarrow i_0 + (3S_M + 3S_N - 2) \cdot ite$ 
4:      $p \leftarrow \alpha + \beta \cdot b(q + 0) + \gamma \cdot b(q + 1)$ 
5:     for  $i \leftarrow 0, (S_M - 1)$  do
6:        $f'_{sub} \leftarrow \text{ROLR}_{b(i+q)}^{i,p}(f_{sub})$ 
7:     end for
8:     for  $j \leftarrow 0, (S_N - 1)$  do
9:        $f'_{sub} \leftarrow \text{ROUD}_{b(j+q+S_M)}^{j,p}(f'_{sub})$ 
10:    end for
11:    for  $k \leftarrow 0, (S_M + S_N - 2)$  do
12:       $f'_{sub} \leftarrow \text{ROUR}_{b(k+q+S_M+S_N)}^{k,p}(f'_{sub})$ 
13:    end for
14:    for  $l \leftarrow (1 - S_N), (S_M - 1)$  do
15:       $f'_{sub} \leftarrow \text{ROUL}_{b(l+q+2 \cdot S_M+3 \cdot S_N-2)}^{l,p}(f'_{sub})$ 
16:    end for
17:  end for
18:   $i_0 \leftarrow i_0 + (3S_M + 3S_N - 2) \cdot no$ 
19:  return ( $f'_{sub}, i_0$ )
20: end function

```

that the Sub_HCIE function actually defines an $S_M \times S_N$ permutation relationship matrix pseudo-randomly controlled by $(3S_M + 3S_N - 2) \cdot no$ bits in the bit sequence $\{b(i)\}$ from i_0 . Based on this function, for an $M \times N$ image $f = [f(i, j)]_{M \times N}$, the four basic parts of HCIE can be briefly described as follows.

- The secret key is the initial condition $x(0)$ and the control parameter μ of the chaotic Logistic map, $f(x) = \mu x(1 - x)$ [18], which is realized in L -bit finite precision.
- Some public parameters: $S_M, S_N, \alpha, \beta, \gamma$ and no , where $\sqrt{M} \leq S_M \leq M, M \bmod S_M = 0, \sqrt{N} \leq S_N \leq N$, and $N \bmod S_N = 0$.
Note: Although $(S_M, S_N, \alpha, \beta, \gamma, no)$ can all be included in the secret key, they are not suitable for such a use due to the following reasons: 1) S_M, S_N are related to M, N ; 2) α, β, γ are related to S_M, S_N (and then related to M, N , too); 3) S_M, S_N can be easily guessed from the mosaic effect of the cipher-image; 4) iteration number no cannot be too large to achieve an acceptable encryption speed.
- The initialization procedure of generating the bit sequence used in the Sub_HCIE function: run the Lo-

gistic map starting from $x(0)$ to generate a chaotic sequence, $\{x(i)\}_{i=0}^{\lceil L_b/8 \rceil - 1}$, and then extract 8 bits following the decimal point of each chaotic state $x(i)$ to yield a bit sequence $\{b(i)\}_{i=0}^{L_b-1}$, where $L_b = \left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right) \cdot (3S_M + 3S_N - 2) \cdot no$; finally, set $i_0 = 0$ to let the Sub_HCIE function run starting from $b(0)$.

- *The two-level hierarchical encryption procedure:*

1) *The high-level encryption – permuting image blocks:* divide the plain-image f into blocks of size $S_M \times S_N$, which compose an $\frac{M}{S_M} \times \frac{N}{S_N}$ block-image

$$P_f = \left[P_f(i, j) \right]_{\frac{M}{S_M} \times \frac{N}{S_N}},$$

where $P_f(i, j)$ is the block of size $S_M \times S_N$ at the position (i, j) . Then, permute the positions of all blocks with the Sub_HCIE function in the following way: a) create a pseudo-image $f_p = [f_p(i, j)]_{S_M \times S_N}$ containing $\left(\frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ non-zero indices of all image blocks in P_f and $\left(S_M \cdot S_N - \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ zero-elements, and permute f_p with the Sub_HCIE function to get a shuffled pseudo-image f_p^* ; b) generate a permuted block-image P_{f^*} from P_f (i.e., permute f blockwise) using the shuffled indices contained in f_p^* . The above high-level encryption procedure can be considered as a permutation of the block-image: $P_f \xrightarrow{f_p^* = \text{Sub_HCIE}(f_p)} P_{f^*}$, where f_p^* actually corresponds to an $\frac{M}{S_M} \times \frac{N}{S_N}$ permutation relationship matrix.

2) *The low-level encryption – permuting pixels in every image block one by one:* for $i = 0 \sim \left(\frac{M}{S_M} - 1\right)$ and $j = 0 \sim \left(\frac{N}{S_N} - 1\right)$, call the Sub_HCIE function to permute each block $P_{f^*}(i, j)$ so as to get the corresponding block of the cipher-image f' :

$$P_{f'}(i, j) = \text{Sub_HCIE}\left(P_{f^*}(i, j)\right).$$

As normalized in [10], any permutation-only encryption algorithm exerting on an object of size $H \times W$ can be represented with a *permutation relationship matrix* of size $H \times W$, denoted by

$$\mathbf{W} = [w(i, j) = (i', j') \in \mathbb{H} \times \mathbb{W}]_{H \times W}, \quad (1)$$

where $\mathbb{H} = \{0, \dots, H - 1\}$, $\mathbb{W} = \{0, \dots, W - 1\}$, and $w(i_1, j_1) \neq w(i_2, j_2)$ for any $(i_1, j_1) \neq (i_2, j_2)$.

In HCIE, a total of $\left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ permutation relationship matrices are involved: 1) one high-level permutation relationship matrix of size $\frac{M}{S_M} \times \frac{N}{S_N}$; 2) $\left(\frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ low-level permutation relationship matrices of size $S_M \times S_N$. With the above-mentioned representation of a permutation-only image encryption algorithm, the secret key $(\mu, x(0))$ of HCIE

is equivalent to the $\left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ permutation relationship matrices for plain-images of the same size. To facilitate the following discussions, we use $\mathbf{W}_0 = [w_0(i, j)]_{\frac{M}{S_M} \times \frac{N}{S_N}}$ to denote the high-level permutation relationship matrix, and use $\{\mathbf{W}_{(i, j)}\}_{i=0, j=0}^{\frac{M}{S_M}-1, \frac{N}{S_N}-1}$ to denote the $\left(\frac{M}{S_M} \times \frac{N}{S_N}\right)$ low-level permutation relationship matrices, where $\mathbf{W}_{(i, j)} = [w_{(i, j)}(i', j')]_{S_M \times S_N}$. Apparently, the $\left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ permutation relationship matrices can be easily transformed to an equivalent permutation relationship matrix of size $M \times N$, $\mathbf{W} = [w(i, j)]_{M \times N}$.

When $S_M = M$ and $S_N = N$ (or $S_M = S_N = 1$), the two hierarchical encryption levels merge into a single one; the $\left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ permutation relationship matrices become one permutation relationship matrix of size $M \times N$, a typical permutation-only image encryption algorithm in which each pixel can be independently permuted to any other position in the whole image by a single $M \times N$ permutation relationship matrix \mathbf{W} .

3. Cryptanalysis of HCIE

3.1. Ciphertext-only attack

In [16], it was claimed that the complexity of brute-force attacks to HCIE is $O(2^{L_b})$, since there are $L_b = \left(1 + \frac{M}{S_M} \cdot \frac{N}{S_N}\right) \cdot (3S_M + 3S_N - 2) \cdot no$ secret chaotic bits in $\{b(i)\}_{i=0}^{L_b-1}$ that are unknown to the attackers. However, this statement is not true due to the following fact: the L_b bits are uniquely determined by the secret key, i.e., the initial condition $x(0)$ and the control parameter μ , which have only $2L$ secret bits. This means that there are only 2^{2L} different chaotic bit sequences.

Now, let us study the real complexity of brute-force attacks. For each pair of guessed values of $x(0)$ and μ , the following operations are needed:

- generating the chaotic bit sequence: there are $L_b/8$ chaotic iterations;
- creating the pseudo-image f_p : the complexity is $S_M \cdot S_N$;
- shuffling the pseudo-image f_p : running the Sub_HCIE function once;
- generating P_{f^*} : the complexity is $M \cdot N$;
- shuffling the partition image P_{f^*} : running the Sub_HCIE function for $\left(\frac{M}{S_M} \cdot \frac{N}{S_N}\right)$ times.

Assume that the computing complexity of the Sub_HCIE function is $(4S_M + 4S_N) \cdot no$. Then, the total complexity of brute-force attacks to HCIE can be estimated

to be $O(2^{2L} \cdot (L_b + M \cdot N))$, which is much smaller than $O(2^{L_b/8})$ when L_b is not too small. Additionally, considering the fact that the Logistic map can exhibit a sufficiently strong chaotic behavior only when μ is close to 4 [19], the above complexity should be even smaller. This analysis shows that the security of HCIE was much over-estimated by the authors in [20, 16], for brute-force attacks.

Observing the hierarchical permutation structure of HCIE, one can see that the histogram of each $S_M \times S_N$ block in the plain-image will keep unchanged during the whole permutation process of HCIE. Due to the strong correlation between neighbouring pixels (and even blocks) of natural images (See [21, Fig. 5]), there exists some correlation between histograms of neighbouring blocks. So, one may determine the relative locations of some blocks in cipher-image by comparing similarity degrees of histograms for every pair of cipher-blocks [22, 23].

3.2. The known-plaintext attack

Since HCIE is a permutation-only image encryption algorithm, given n known plain-images $f_1 \sim f_n$ of size $M \times N$ and the corresponding cipher-images $f'_1 \sim f'_n$, one can simply call the `Get_Permutation_Matrix` function defined in [10, Sec. 3.1] or its enhanced version in [11, Sec. 4] with the input parameter $(f_1 \sim f_n, f'_1 \sim f'_n, M, N)$ to estimate an $M \times N$ permutation relationship matrix \mathbf{W} , which is equivalent to the $(1 + \frac{M}{S_M} \cdot \frac{N}{S_N})$ smaller permutation relationship matrices. However, if the hierarchical structure of HCIE is considered, the known-plaintext attack may be quicker and the estimation will be more effective. Thus, the following hierarchical procedure of known-plaintext attacks to HCIE is suggested¹:

- *Reconstruct the high-level permutation relationship matrix \mathbf{W}_0 :* 1) for $i = 0 \sim (\frac{M}{S_M} - 1)$ and $j = 0 \sim (\frac{N}{S_N} - 1)$: calculate the mean values of the $2n$ blocks $P_{f_1}(i, j) \sim P_{f_n}(i, j)$, $P_{f'_1}(i, j) \sim P_{f'_n}(i, j)$ and denote them by $\overline{P_{f_1}}(i, j) \sim \overline{P_{f_n}}(i, j)$ and $\overline{P_{f'_1}}(i, j) \sim \overline{P_{f'_n}}(i, j)$;
- 2) generate $2n$ images $\overline{P_{f_1}} \sim \overline{P_{f_n}}$ and $\overline{P_{f'_1}} \sim \overline{P_{f'_n}}$ of size $\frac{M}{S_M} \times \frac{N}{S_N}$ as follows: $\forall m = 1 \sim n$,

$$\overline{P_{f_m}} = \left[\overline{P_{f_m}}(i, j) \right]_{\frac{M}{S_M} \times \frac{N}{S_N}} \quad (2)$$

and

$$\overline{P_{f'_m}} = \left[\overline{P_{f'_m}}(i, j) \right]_{\frac{M}{S_M} \times \frac{N}{S_N}}, \quad (3)$$

¹For HCIE, the permutation relationship matrices also depend on the values of the public parameters. To simplify the following description, without loss of generality, it is assumed that all public parameters are fixed for all known plain-images.

and call the `Get_Permutation_Matrix` function with the input parameters

$$\left(\overline{P_{f_1}} \sim \overline{P_{f_n}}, \overline{P_{f'_1}} \sim \overline{P_{f'_n}}, \frac{M}{S_M}, \frac{N}{S_N} \right)$$

to get an estimated permutation relationship matrix $\widehat{\mathbf{W}}_0 = [\widehat{w}_0(i, j)]_{\frac{M}{S_M} \times \frac{N}{S_N}}$ and its inverse $\widehat{\mathbf{W}}_0^{-1} = [\widehat{w}_0^{-1}(i, j)]_{\frac{M}{S_M} \times \frac{N}{S_N}}$.

3) *Reconstruct the $(\frac{M}{S_M} \cdot \frac{N}{S_N})$ low-level permutation relationship matrices $\{\mathbf{W}_{(i,j)}\}_{i=0, j=0}^{\frac{M}{S_M}-1, \frac{N}{S_N}-1}$:*

- for $i = 0 \sim (\frac{M}{S_M} - 1)$ and $j = 0 \sim (\frac{N}{S_N} - 1)$, call the `Get_Permutation_Matrix` function with the input parameters $(P_{f_1}(i, j) \sim P_{f_n}(i, j), P_{f'_1}(i', j') \sim P_{f'_n}(i', j'), S_M, S_N)$, where $(i', j') = \mathbf{W}_0(i, j)$, to determine an estimated permutation relationship matrix $\widehat{\mathbf{W}}_{(i,j)}$ and its inverse $\widehat{\mathbf{W}}_{(i,j)}^{-1}$.

With the $(1 + \frac{M}{S_M} \cdot \frac{N}{S_N})$ inverse matrices \mathbf{W}_0^{-1} and $\{\widehat{\mathbf{W}}_{(i,j)}^{-1}\}_{i=0, j=0}^{\frac{M}{S_M}-1, \frac{N}{S_N}-1}$, one can decrypt a new cipher-image f'_{n+1} with `Dermutation` function given in Algorithm 2 to get an estimated plain-image f_{n+1}^* :

Algorithm 2 The function `Dermutation`

```

1: function DERMUTATION( $\mathbf{W}_0^{-1}, \{\widehat{\mathbf{W}}_{(i,j)}^{-1}\}_{i=0, j=0}^{\frac{M}{S_M}-1, \frac{N}{S_N}-1}, f'_{n+1}$ )
2:   for  $i \leftarrow 0, (M/S_M) - 1$  do
3:     for  $j \leftarrow 0, (N/S_N) - 1$  do
4:        $f_{temp} \leftarrow P_{f'_{n+1}}(\mathbf{W}_0^{-1}(i, j))$ 
5:       for  $ii \leftarrow 0, S_M - 1$  do
6:         for  $jj \leftarrow 0, S_N - 1$  do
7:            $f_{temp}(ii, jj) \leftarrow f_{temp}(\mathbf{W}_{(i,j)}^{-1}(ii, jj))$ 
8:            $P_{f'_{n+1}}(i, j) \leftarrow f_{temp}$ 
9:         end for
10:      end for
11:    end for
12:  end for
13:  return  $f_{n+1}^*$ 
14: end function

```

In fact, in the above procedure, any measure keeping invariant in the block permutations can be used instead of the mean value. A typical measure is the histogram of each $S_M \times S_N$ block. Although the mean value is less precise than the histogram, it works well in most cases and is effective to reduce the time complexity. When T and $S_M \times S_N$ are both very small, the efficiency of the mean value will become low, in this case the histogram or the array of all

pixel values can be used as a replacement. As for an image of size $H \times W$ and T possible value levels, the number of different histogram is

$$n_h = \sum_{i=1}^{\min(T, H \cdot W)} \binom{T}{i} \cdot \binom{H \cdot W}{i-1}. \quad (4)$$

As n_h is a huge number for a non-tiny image and histogram is sensitive to the change of pixel value, it is easier to get the high-level permutation relationship matrix \mathbf{W}_0 than the low-level permutation relationship matrices.

Finally, let us see whether the hierarchical structure used in HCIE is helpful for enhancing the security against the known-plaintext attack to the common permutation image ciphers. As discussed in [10, Sec. 3.1], $n \geq \lceil \log_T(2(M \cdot N - 1)) \rceil$ known plain-images are needed to achieve an acceptable breaking performance. Since the hierarchical structure makes it possible for an attacker to work on permutation relationship matrices of size $S_M \times S_N$ or $\frac{M}{S_M} \times \frac{N}{S_N}$ (both smaller than $M \times N$), it is obvious that for HCIE the number of required known plain-images will be smaller than $\lceil \log_T(2(M \cdot N - 1)) \rceil$. As the permutation relationship matrix \mathbf{W}_0 can be recovered in a very high probability with one or two known plain-images, one can conclude as follows: the smaller the $(S_M \times S_N)$ is, the smaller the n is. Also, the attack complexity will become lower, since the number of used plaintexts is reduced. In this sense, hierarchical permutation-only image ciphers are less secure than non-hierarchical ones, which discourages the use of HCIE.

3.3. Experimental results on known-plaintext attack

To verify the decryption performance of the above-discussed known-plaintext attack to HCIE, some experiments are performed using the six 256×256 test images with 256 gray scales shown in Fig. 1. Assume that the first $n = 1 \sim 5$ test images are known to an attacker, the cipher-image of the last test image is decrypted with the estimated permutation relationship matrices, to evaluate the breaking performance.

In the experiments, three different configurations of HCIE are used: $S_M = S_N = 256$, $S_M = S_N = 32$, $S_M = S_N = 16$. As mentioned above, the configuration of $S_M = S_N = 256$ corresponds to general permutation-only image ciphers working in the spatial domain (without using hierarchical structures). As shown in [10, Sec. 4], three known plain-images are always enough to achieve a good breaking performance, and an almost perfect breaking performance can be achieved with four plain-images, where the public parameters are $\alpha = 6, \beta = 3, \gamma = 3$ and $no = 9$.

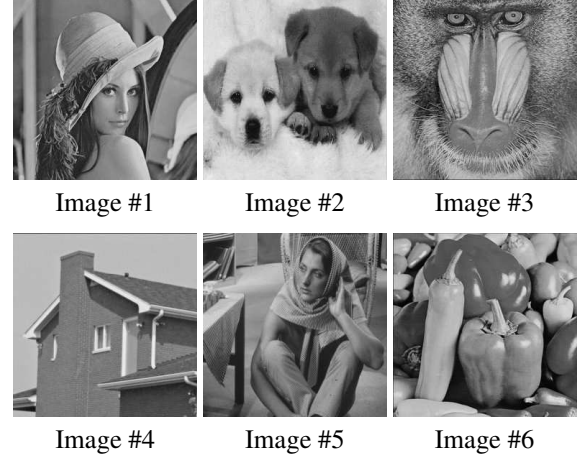


Figure 1: The six 256×256 test images used in the experiments.

3.3.1. The experimental results with $S_M = S_N = 32$

The public parameters are set as follows: $\alpha = 4, \beta = 2, \gamma = 1$ and $no = 2$. The cipher-images of the six test images are all shown in Fig. 2. When the first $n = 1 \sim 5$ test images are known to the attacker, the obtained five decrypted images of the sixth cipher-image are shown in Fig. 3. As can be seen, one known plain-image cannot reveal much useful visual information, but two is enough to obtain a good performance.

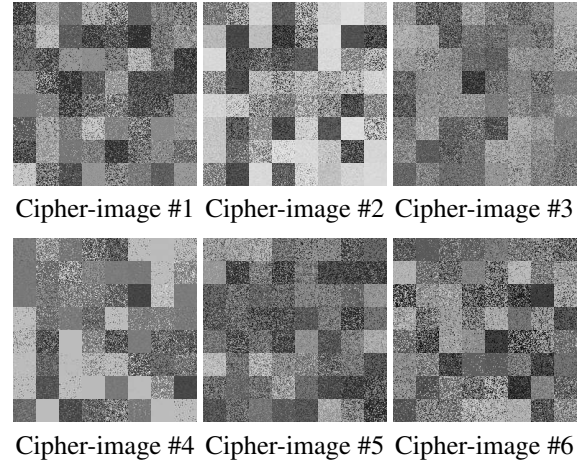


Figure 2: The cipher-images of the six 256×256 test images, when $S_M = S_N = 32$.

3.3.2. The experimental results with $S_M = S_N = 16$

The public parameters are $\alpha = 4, \beta = 2, \gamma = 1$ and $no = 2$. The cipher-images of the six test images are all shown in Fig. 4. When the first $n = 1 \sim 5$ test images are known to the attacker, the five decrypt images of the sixth cipher-image are shown in Fig. 5. As can be seen,

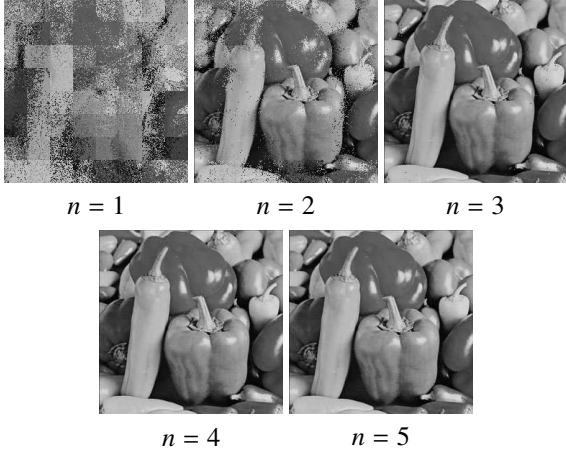


Figure 3: The decrypted image of Cipher-Image #6 when the first n test images are known to the attacker, when $S_M = S_N = 32$.

even one known plain-image can reveal a rough view of the plain-image, and two is enough to obtain a nearly-perfect recovery.

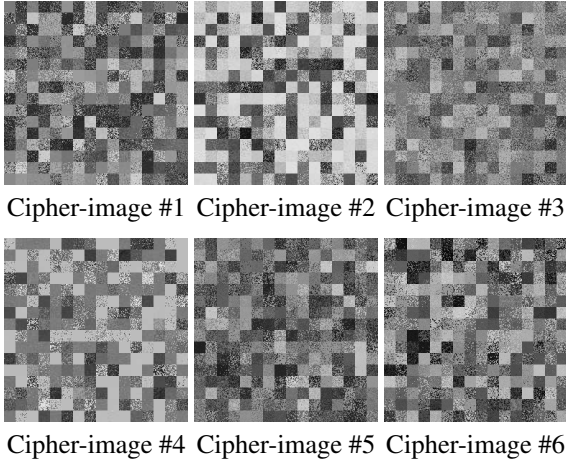


Figure 4: The cipher-images of the six 256×256 test images, when $S_M = S_N = 16$.

Now, let us give a performance comparison of the known-plaintext attack to HCIE with the above three different configurations. Figure 6a) shows the quantitative relationship between the number of known plain-images and the decryption quality (represented by the decryption error ratio). It can be seen that three known plain-images are enough for all three configurations to achieve an acceptable breaking performance, and two can reveal quite a lot of pixels (which means that most significant visual information is revealed). Also, it is found that the breaking performance is dependent on the configuration: when $S_M = S_N = 16$, the best performance is achieved, which coincides with the

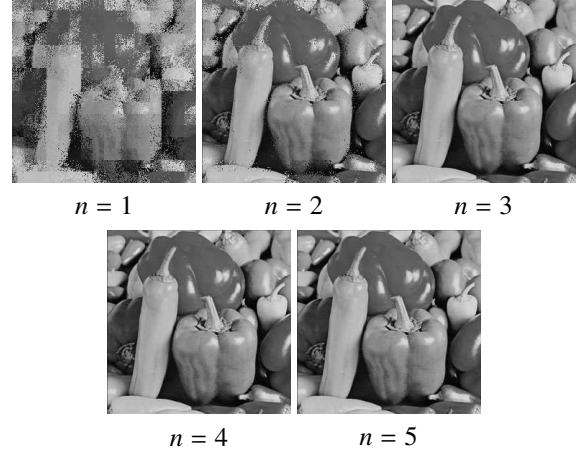


Figure 5: The decrypted images of Cipher-Image #6 when the first n test images are known to an attacker, when $S_M = S_N = 16$.

previous analysis.

Figure 6b) shows the average cardinality of the elements in \bar{W} , which is an indicator of the probability of getting correct permutation elements in \bar{W} and an indicator of the time complexity as analyzed above. Comparing Figures 6a) and 6b), one can see that the occurrence probability of decryption errors has a good correspondence with the average cardinality, where the correctness of the uniquely-determined permutation relationship matrix was obtained by some chosen plain-images and the corresponding cipher-images.

From the above comparison, it is concluded that the security of HCIE with a hierarchical structure is even weaker than the security of general permutation-only image encryption algorithms without hierarchical structures: when $S_M = S_N = 32$ and $S_M = S_N = 16$, two known plain-images are enough to achieve an acceptable breaking performance; while when $S_M = S_N = 256$, the breaking performance with two known plain-images is not satisfactory, thus three plain-images are needed to achieve an acceptable performance. Overall, from the viewpoint of security against known/chosen-plaintext attacks, the hierarchical idea proposed in HCIE has no technical merits.

3.4. Chosen-plaintext attack

To discover an equivalent secret key of a common permutation-only image encryption scheme under the scenario of chosen-plaintext attack, one can construct a “composition plain-text”, whose every element is different from each other [24, Sec. 5.1]. To satisfy the requirement, the number of the bit planes of the special chosen-text should be not smaller than $\lceil \log_2(M \cdot N) \rceil$. So, the number of required chosen-images is

$$n = \lceil \lceil \log_2(M \cdot N) \rceil / \lceil \log_2(T) \rceil \rceil.$$

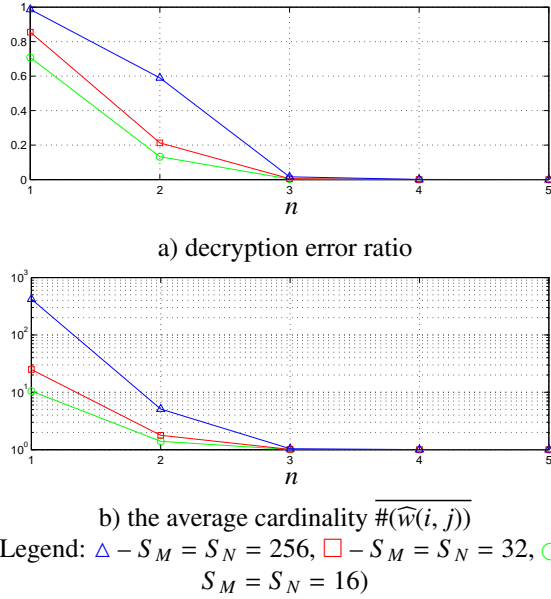


Figure 6: A performance comparison of the known-plaintext attack to HCIE.

In general, T is a power of 2 in the digital domain, hence $\log_2(T)$ is an integer. To this end, one has

$$n = \lceil \log_2(M \cdot N) / \log_2(T) \rceil = \lceil \log_T(M \cdot N) \rceil \quad (5)$$

by referring to [25, Theorem 3.10].

Similarly to the known-plaintext attack, the use of a hierarchical structure in HCIE can also make the construction of chosen plain-images easier. Accordingly, an attacker can work hierarchically to construct n chosen plain-images, f_1, \dots, f_n , as follows:

- *high-level*: $\bar{P}_{f_1} \sim \bar{P}_{f_n}$, which are defined in Eq. (2), compose an orthogonal image set;
- *low-level*: $\forall(i, j), P_{f_1}(i, j) \sim P_{f_n}(i, j)$ compose an orthogonal image set.

In this case, the minimal number of required chosen plain-images becomes

$$\begin{aligned} n &= \max(\lceil \log_T(S_M \cdot S_N) \rceil, \lceil \log_T(K) \rceil) \\ &\leq \lceil \log_T(M \cdot N) \rceil, \end{aligned} \quad (6)$$

where $K = \frac{M}{S_M} \cdot \frac{N}{S_N}$. The above equality holds if and only if the hierarchical encryption structure is disabled, i.e., when $K \in \{1, M \cdot N\}$.

As the $(1 + \frac{M}{S_M} \cdot \frac{N}{S_N})$ permutation relationship matrices of HCIE are uniquely determined by the bit sequence $\{b(i)\}_{i=0}^{L_b-1}$ and the public parameters, one may recover reversely some consecutive bits of $\{b(i)\}_{i=0}^{L_b-1}$ [26, Sec. 3.3.6]. Furthermore, one can derive the secret key $x(0)$ and μ following the approach given in [21, Sec.3.3.2].

4. Conclusion

Specific security performance of a typical permutation-only encryption algorithm, called HCIE, against ciphertext-only attack and known/chosen-plaintext attacks has been studied in detail. It is found that the capability of HCIE against the former attack was over-estimated much and hierarchical permutation-only image encryption algorithms such as HCIE are less secure than normal permutation-only ones without using hierarchical encryption structures. This work effectively demonstrates that the size of the real permutation domain of a permutation algorithm should be as large as possible in order to reach the best performance. As permutation operation alone cannot provide high level of security, it should be combined with other value substitution functions.

Acknowledgement

This research was supported by the Distinguished Young Scholar Program of the Hunan Provincial Natural Science Foundation of China (No. 2015JJ1013). Some parts of Sec. 3 were completed with the help of Dr. Shujun Li, from Surrey University, UK.

References

- [1] S. Li, Analyses and new designs of digital chaotic ciphers, Ph.D. thesis, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China, available online at <http://www.hooklee.com/pub.html> (June 2003).
- [2] G. Chen, Y. Mao, C. K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, *Chaos, Solitons & Fractals* 21 (3) (2004) 749–761.
- [3] G. Álvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, *International Journal of Bifurcation and Chaos* 16 (8) (2006) 2129–2151.
- [4] J. P. Crutchfield, J. D. Farmer, N. H. Packard, R. S. Shaw, *Chaos*, Scientific American 255 (12) (1986) 46–57. doi:10.1038/scientificamerican1286-46.
- [5] Y. Matias, A. Shamir, A video scrambling technique based on space filling curve (extended abstract), in: *Advances in Cryptology – Crypto'87*, Lecture Notes in Computer Science, volume 293, 1987, pp. 398–417.
- [6] W. Zeng, S. Lei, Efficient frequency domain selective scrambling of digital video, *IEEE Transactions on Multimedia* 5 (1) (2003) 118–129.
- [7] J.-K. Jan, Y.-M. Tseng, On the security of image encryption method, *Information Processing Letters* 60 (5) (1996) 261–265.
- [8] C.-C. Chang, T.-X. Yu, Cryptanalysis of an encryption scheme for binary images, *Pattern Recognition Letters* 23 (14) (2002) 1847–1852.
- [9] W. Li, Y. Yan, N. Yu, Breaking row-column shuffle based image cipher, in: *Proceedings of ACM International Conference on Multimedia*, 2012, p. art. no. 6011939. doi:10.1145/2393347.2396392.
- [10] S. Li, C. Li, G. Chen, N. G. Bourbakis, K.-T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, *Signal Processing: Image Communication* 23 (3) (2008) 212–223, <https://eprint.iacr.org/2004/374.pdf>.

- [11] C. Li, K.-T. Lo, Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, *Signal Processing* 91 (4) (2011) 949–954.
- [12] S. Li, C. Li, K.-T. Lo, G. Chen, Cryptanalysis of an image scrambling scheme without bandwidth expansion, *IEEE Transactions on Circuits and Systems for Video Technology* 18 (3) (2008) 338–349.
- [13] H. Sohn, W. D. Neve, Y. M. Ro, Privacy protection in video surveillance systems: analysis of subband-adaptive scrambling in JPEG XR, *IEEE Transactions on Circuits and Systems for Video Technology* 21 (2) (2011) 170–177.
- [14] Y. Wu, Y. Zhou, S. Agaian, J. P. Noonana, A symmetric image cipher using wave perturbations, *Signal Processing* 102 (2014) 122–131.
- [15] H. Zheng, S. Yu, J. Lu, Design and ARM platform-based realization of digital color image encryption and decryption via single state variable feedback control, *International Journal of Bifurcation and Chaos* 24 (4) (2014) art. no. 1450049.
- [16] J.-C. Yen, J.-I. Guo, Efficient hierarchical chaotic image encryption algorithm and its VLSI realisation, *IEE Proc. – Vis. Image Signal Process.* 147 (2) (2000) 167–175.
- [17] R. E. Korf, Finding optimal solutions to Rubik’s cube using pattern databases, in: *Proceedings of AAAI-97*, 1997, pp. 700–705, <https://www.aaai.org/Papers/AAAI/1997/AAAI97-109.pdf>.
- [18] R. L. Devaney, *An introduction to chaotic dynamical systems*, Westview Press, 2003.
- [19] C. Li, T. Xie, Q. Liu, G. Cheng, Cryptanalyzing image encryption using chaotic logistic map, *Nonlinear Dynamics* 78 (2) (2014) 1545–1551.
- [20] J.-I. Guo, J.-C. Yen, J.-C. Yeh, The design and realization of a new chaotic image encryption algorithm, in: *Proc. 1999 International Symposium on Communications*, 1999, pp. 210–214.
- [21] S. Li, C. Li, G. Chen, X. Mou, Cryptanalysis of the RCES/RSES image encryption scheme, *Journal of Systems and Software* 81 (7) (2008) 1130–1143.
- [22] H. Li, Y. Zheng, S. Zhang, J. Cheng, Solving a special type of jigsaw puzzles: Banknote reconstruction from a large number of fragments, *IEEE Transactions on Multimedia* 16 (2) (2014) 571–578.
- [23] H. Ling, K. Okada, An efficient earth mover’s distance algorithm for robust histogram comparison, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (5) (2007) 840–853.
- [24] C. Li, S. Li, G. Chen, G. Chen, L. Hu, Cryptanalysis of a new signal security system for multimedia data transmission, *EURASIP Journal on Applied Signal Processing* 2005 (8) (2005) 1277–1288.
- [25] R. L. Graham, D. E. Knuth, O. Patashnik, *Concrete Mathematics*, Addison-Wesley, 1989.
- [26] C. Li, S. Li, K.-T. Lo, K. Kyamakya, A differential cryptanalysis of Yen-Chen-Wu multimedia cryptography system, *Journal of Systems and Software* 83 (8) (2010) 1443–1452.